Project 4 Task 2 – Tone Analyser App

Description:

My application takes an input text string from the user in the android app, and uses it to fetch and display the analysed tone scores for the text from IBM Watson API.

Here is how my application meets the task requirements:

1. Log useful information

8 pieces of information is logged for each request/reply with the mobile phone. It includes information about the request from the mobile phone, information about the request and reply to the 3rd party API, and information about the reply to the mobile phone.

It includes fields like:

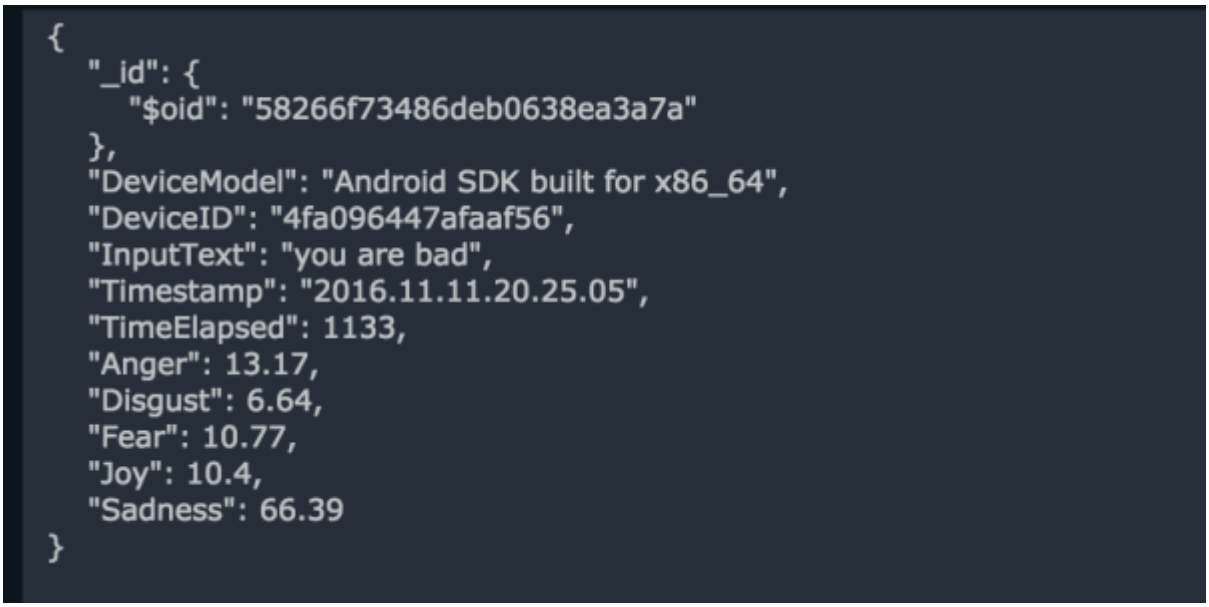
- DeviceModel
- DeviceID
- InputText
- Timestamp
- TimeElapsed
- Anger
- Disgust
- Fear
- Joy
- Sadness

2. Store the log information in a database

The web service can connect, store, and retrieve information from a MongoDB database named as: "**toneanalysis**" in the cloud.

The format it is stored in MongoDB database is:


```
{
   "_id": {
      "$oid": "5826668a486deb06380cc652"
   },
   "DeviceModel": "Android SDK built for x86_64",
   "DeviceID": "4fa096447afaaf56",
   "InputText": "will you marry me",
   "Timestamp": "2016.11.11.19.47.04",
   "TimeElapsed": 2239,
   "Anger": 9.38,
   "Disgust": 1.18,
   "Fear": 15.88,
   "Joy": 53.09,
   "Sadness": 25.99
}
```

StoreRecord method requests the MongoClient using the database username and password on Mlab and storing the collection one by one.

MongoClientURI :

"mongodb://nandita:Kitkat16#@ds053216.mlab.com:53216/toneanalysis"

3. Display operations analytics and full logs on a web-based dashboard

3.1. A unique Heroku URL which addresses a web interface dashboard for the web service:

https://nameless-thicket-87312.herokuapp.com/getAnalysis

3.2. The dashboard displays at least 5 interesting operations analytics.

Detailed analytics of your input texts are:

Top 5 *Anger* Tone Texts:

Input Text	Anger(%)	Disgust(%)	Fear(%)	Joy(%)	Sadness(%)
angry face	96.64	1.13	3.11	0.08	0.8
angry	90.0	80.0	0.0	0.0	0.0
i hate you	55.26	5.72	12.13	2.51	30.81
alvin and chip	22.59	11.65	13.16	43.67	15.68
i want to die	18.39	12.58	4.1	1.11	70.8

Top 5 *Disgust* Tone Texts:

Input Text	Anger(%)	Disgust(%)	Fear(%)	Joy(%)	Sadness(%)
angry	90.0	80.0	0.0	0.0	0.0
hell on you	10.13	37.81	19.55	5.52	36.04
what is your name	7.03	14.76	46.98	16.33	21.89
Nandita is a good girl	4.19	13.63	2.54	72.93	13.12
i will kill you	14.27	12.95	53.75	2.49	25.14

Top 5 *Fear* Tone Texts:

Input Text	Anger(%)	Disgust(%)	Fear(%)	Joy(%)	Sadness(%)
this is scary	4.01	4.36	88.17	0.43	7.74
i will kill you	14.27	12.95	53.75	2.49	25.14
what is your name	7.03	14.76	46.98	16.33	21.89
i want to live	7.89	3.78	25.95	47.44	20.74
i have a doubt	7.25	9.95	25.78	15.92	47.8

Top 5 *Joy* Tone Texts:

Input Text	Anger(%)	Disgust(%)	Fear(%)	Joy(%)	Sadness(%)
:)	0.0	0.0	0.0	100.0	0.0
happy smile	0.16	0.4	0.06	99.15	0.62
you have a lovely dress	4.47	2.94	3.35	80.37	12.98
colorful dress	2.85	10.22	9.54	78.1	6.11
are you crazy	9.74	0.92	12.37	73.11	10.37

Top 5 *Sadness* Tone Texts:

Input Text	Anger(%)	Disgust(%)	Fear(%)	Joy(%)	Sadness(%)
:(	0.0	0.0	0.0	0.0	100.0
sad face	0.59	1.13	2.0	1.09	96.63
i miss you	0.68	1.96	0.5	1.78	96.02
why did u do	17.18	8.22	10.11	4.07	71.21
why did u do	17.18	8.22	10.11	4.07	71.21

Few more insights & analysis

1. Text having the highest Joy tone in it: *:)* Score = 100.0%
2. Text having the highest Anger tone in it: ***angry face*** Score = 96.64%
3. Total texts analyzed: **32**
4. Highest Latency is : ***2335.0 msecs*** for: ***Nandita is a good girl*** input text

3.3. The dashboard displays the full logs.

Displaying all the records stored in database

Database Dump:

Device Model	Device ID	Input Text	Timestamp(yyyy.mm.dd.hh.mm.sec)	Latency(mSec)	Anger(%)	Disgust(%)	Fear(%)	Joy(%)	Sadness(%)
Android SDK built for x86_64	4fa096447afaaf56	will you marry me	2016.11.11.19.47.04	2239	9.38	1.18	15.88	53.09	25.99
Android SDK built for x86_64	4fa096447afaaf56	you are bad	2016.11.11.20.25.05	1133	13.17	6.64	10.77	10.4	66.39
Android SDK built for x86_64	4fa096447afaaf56	hell on you	2016.11.11.21.19.34	1872	10.13	37.81	19.55	5.52	36.04
Android SDK built for x86_64	4fa096447afaaf56	alvin and chip	2016.11.11.21.24.31	2000	22.59	11.65	13.16	43.67	15.68
Android SDK built for x86_64	4fa096447afaaf56	:(	2016.11.11.21.50.39	1456	0.0	0.0	0.0	0.0	100.0
Android SDK built for x86_64	4fa096447afaaf56	iseuosgsgsejbjiebitewbitbiw	2016.11.11.21.50.48	1090	0.0	0.0	0.0	0.0	0.0
Android SDK built for x86_64	296b7e9386033b95	i have a doubt	2016.11.12.02.09.52	1817	7.25	9.95	25.78	15.92	47.8
Android SDK built for x86_64	296b7e9386033b95	this is scary	2016.11.12.02.10.08	788	4.01	4.36	88.17	0.43	7.74
Android SDK built for x86_64	296b7e9386033b95	colorful dress	2016.11.12.02.10.26	759	2.85	10.22	9.54	78.1	6.11
Android SDK built for x86_64	296b7e9386033b95	you have a lovely dress	2016.11.12.03.24.40	1696	4.47	2.94	3.35	80.37	12.98
Android SDK built for x86_64	296b7e9386033b95	Nandita is a good girl	2016.11.12.04.16.17	2335	4.19	13.63	2.54	72.93	13.12
Android SDK built for x86_64	296b7e9386033b95	what is your name	2016.11.11.19.29.16	1213	7.03	14.76	46.98	16.33	21.89
Android SDK built for x86_64	296b7e9386033b95	go to helloworld	2016.11.11.19.38.19	2159	11.61	3.29	23.72	13.78	54.87
Android SDK built for x86_64	296b7e9386033b95	i love you	2016.11.11.19.39.46	681	3.37	0.2	2.95	64.73	33.52
Android SDK built for x86_64	296b7e9386033b95	i hate you	2016.11.11.19.44.55	1811	55.26	5.72	12.13	2.51	30.81
Android SDK built for x86_64	296b7e9386033b95	i will kill you	2016.11.11.19.45.20	684	14.27	12.95	53.75	2.49	25.14
Android SDK built for x86_64	4fa096447afaaf56	why did u do	2016.11.11.20.46.32	1751	17.18	8.22	10.11	4.07	71.21
Android SDK built for x86_64	4fa096447afaaf56	i want to die	2016.11.12.00.38.14	762	18.39	12.58	4.1	1.11	70.8
Android SDK built for x86_64	4fa096447afaaf56	i miss you	2016.11.12.00.55.12	1720	0.68	1.96	0.5	1.78	96.02
Android SDK built for x86_64	296b7e9386033b95	you are charming	2016.11.12.02.07.10	814	8.15	9.43	15.73	57.09	14.44

4. Deploy the web service to Heroku

Deployed the web service to Heroku. This web service has all the functionality of Task 1 but with the additional logging, database, and dashboard analytics functions.

Dashboard URL: https://nameless-thicket-87312.herokuapp.com/getAnalysis

5. Implement a web application, deployed to Heroku

The URL of my web service deployed to Heroku is:

Heroku Server URL: "https://nameless-thicket-87312.herokuapp.com//ToneAnalysisServer/" + analyzeText;

The project directory name is Project4Task2.

5.1. Using an HttpServlet to implement a simple API

In my web app project:
Model:

- MongoDBStore.java
- FetchResponse.java
- JsonOperations.java
- RecordRetrieve.java

Controller: ToneAnalysisServer.java
View: index.jsp